

Lógica para Semáforo Inteligente Baseado na Mineração de Dados por Algoritmo Genético Transgênico

Lucas Costa Cunha¹, Luciana Aparecida Ferreira¹, Fabrício Alves Rodrigues²,
Wanderley de Souza Alencar¹

¹Curso de Ciência da Computação - Universidade Federal de Goiás/Jataí (UFG)
BR364 Km 192, Setor Industrial, Jataí - GO - Brasil

²Pós-Graduação em Ciência da Computação - Universidade Federal de São Carlos (UFSCar)
Rodovia Washington Luís, km 235 - SP-310 São Carlos - São Paulo - Brasil

l.cunha@hotmail.com, fabricio1989@gmail.com

Abstract. *In large and medium size cities there is a known problem by traffic engineers, traffic flow problem. In computer science this problem is seen in the area of computer networks when there is a congestion somewhere in the network. This article proposes a solution for traffic flow problem using heuristics of Artificial Intelligence, Genetic Algorithms, Data Mining to make a pattern and get specific rules which will be used on the rules fuzzy, for the automation of a traffic light to traffic flow at a certain passage, when there is a need for change to see as this environment.*

Resumo. *Em grandes e medias cidades há um problema já conhecido pelos engenheiros de tráfego, o problema de fluxo de tráfego. Em ciência da computação esse problema é visto na área de redes de computadores quando existe um congestionamento em algum ponto da rede. Este artigo propõe uma solução para o problema de fluxo de tráfego utilizando de heurísticas da inteligência Artificial, Algoritmos Genéticos, Mineração de Dados para criar um padrão e estabelecer regras específicas que serão utilizadas nas regras Fuzzy, para a automação de um Semáforo para que o tráfego flua, em determinado trecho, quando existir a necessidade da mudança ao ver como está o ambiente.*

1. Introdução

Os esforços para a modelagem de problemas de ET (Engenharia de Trânsito) tem se tornado uma alternativa promissora. Problemas de ET, um dos problemas é o fluxo de tráfego, são formulados como problemas de programação linear inteira, ou seja, se tem uma função de otimização e suas restrições para serem analisadas é do tipo NP-completo. A solução destes problemas pode ser obtida em tempo hábil, através de métodos aproximativos [Dias 2004].

Para resolver o problema de fluxo de tráfego muitos pesquisadores tem estudado formas de mudar o tempo do semáforo e assim melhorar um pouco o problema do fluxo em um determinado trecho da malha de tráfego urbano, nos últimos anos tem se tornado um grande desafio a otimização dos tempos do semáforo [Lämmer and Helbing 2008]. O trabalho de Tonguz et al. [Tonguz et al. 2009] apresentou uma nova abordagem de autômatos celulares para a construção de um modelo de tráfego urbano e assim encontrou algumas características dos padrões globais de tráfego nas áreas estudadas e como

resultados mostrou que controles como a duração do ciclo, *split* verde, e coordenação de semáforos mostraram bons resultados para a dinâmica de tráfego.

Já nos trabalhos [Karakuzu and Demirci 2010, Yang et al. 2012, Askerzade and Mahmood 2010] é proposto um semáforo com um controle inteligente para mudanças de fases do sinal, vermelho e verde, utilizando lógica *fuzzy* com objetivo de diminuir o problema do fluxo de tráfego. Para validar os resultados foi desenvolvido um simulador para representar o problema de fluxo com semáforos sincronizados e usando a lógica *fuzzy*.

Com o aumento significativo do uso dos automóveis e a expansão do sistema de trânsito surgiu o estudo de como encontrar características do tráfego, envolvendo modelos matemáticos [Prado 2010]. Neste trabalho será utilizado algoritmo genético, que é um método heurístico que busca os melhores resultados criado por Holland que faz uso do processo evolutivo descrito por Darwin, para descrever uma característica, ou também um padrão, para o tráfego.

O Algoritmo Genético pode ser utilizado para fazer a mineração dos dados como mostra os trabalhos [Amaral 2007, Bevilaqua et al. 2011, Minaei-Bidgoli and Punch 2003, Shah and Kusiak 2007, Cox 2005]. A mineração de dados foi desenvolvida na década de 80 para encontrar algum tipo de padrão em grandes base de dados e é muito utilizado também para geração de novos conhecimentos, neste trabalho os resultados da mineração de dados serão utilizados na lógica *Fuzzy*, que é uma lógica criada por Zadeh que utiliza grau de pertinência, entre 0-1, dos dados analisados e desta forma criar regras difusas para fazer inferência para alcançar a solução do problema.

O Objetivo deste trabalho é mostrar a eficiência do Algoritmo Genético Transgênico, descrito por [Amaral and Jr. 2011], para minerar dados, pois na bibliografia pesquisada não foi encontrado nenhum trabalho que propôs minerar uma base de dados de tráfego para ser criada uma regra difusa, também mostrar a eficiência da lógica difusa para realizar controle de sinal do semáforo e para diferenciar de outros trabalhos será utilizado a mineração de dados em uma base de dados que representa o fluxo de tráfego de um trecho de Jataí, para que a regra *fuzzy* realmente represente o comportamento do tráfego.

Para mostrarmos a eficiência da lógica *Fuzzy* será utilizado o seguinte estudo de caso: Em um trecho de trânsito T1 há um Semáforo que coordena o tráfego daquele trecho, pesquisadores da engenharia de tráfego analisaram o fluxo do trecho T1 por três anos, levando em conta a data, os horários, se houve ocorrência de chuva, quantidade de fluxo naquele período de tempo e se houve ou não congestionamento.

Este estudo de caso tem como base dados recolhidos na cidade de Jataí no trecho da avenida Goiás com a avenida Brasil e será utilizado para mostrar que os métodos de IA (Inteligência Artificial) e as regras difusa podem ajudar o semáforo a tomar uma decisão sobre como irá funcionar o fluxo naquele trecho, levando em conta a hora e a quantidade de fluxo naquele trecho estudado.

2. Referencial Teórico

O estudo científico do fluxo de tráfego teve seu início na década de trinta aplicando a teoria de probabilidade para descrever o comportamento do tráfego na estrada e com modelos

de estudos que relacionavam o volume, a velocidade e a investigação do desempenho do tráfego nos cruzamentos [Prado 2010].

O fluxo do tráfego, com observação de dados reais, é classificado como sendo um efeito espaço-temporal: existe a ocorrência de congestionamento no espaço e no tempo dentro de uma rede de tráfego. Estes padrões empíricos de via congestionada exibem algumas características complexas. Por esta razão, a complexidade da gestão do tráfego está associada com esta variedade de seus padrões, bem como com a necessidade na otimização. Esta otimização pode resolver o problema de congestionamento ou, se isto não é possível alcançar, a minimização da influência de uma situação de via congestionada [Kerner 2009].

Para manipular dados complexos que podem estar associados a um padrão é utilizada a mineração de dados. Em meados de 1980, a Mineração de Dados (MD) (ou também chamada de *Data mining*) foi criada por diversos pesquisadores e grandes empresas que tinham como objetivo criar métodos que conseguissem resolver os desafios descritos acima e para analisar os dados que estavam sendo gerados.

A MD se baseia em abordagens como amostragem, estimativa, teste de hipóteses, algoritmos de busca, teorias da inteligência artificial, reconhecimento de padrões e aprendizado de máquina. A MD atraiu vários pesquisadores de áreas diversas todos com o mesmo objetivo de buscar bons resultados a fim de lidar com o crescimento dos bancos de dados e com sua alta dimensionalidade [Tan et al. 2009].

A mineração de dados é referente a extração de conhecimento dentro de grandes conjuntos de dados, também é associada ao KDD (*Knowledge Discovery Database*), a MD apresenta ao usuário padrões encontrados na base de dados, que muitas das vezes o usuário não encontraria facilmente [Dantas et al. 2008].

Existem vários métodos para fazer a mineração de dados, no *software* Weka [Homes et al. 1994] pode-se encontrar muitos tipos de mineradores, neste artigo será utilizado o algoritmo genético (AG), como minerador de dados.

Os Algoritmos Genéticos são algoritmos de busca, que utilizam o processo da seleção natural proposto por Charles Darwin, os AGs foram inicialmente propostos por John Holland [Holland 1975]. Os AGs são bastante utilizados em problemas que dado um conjunto de elementos ou indivíduos se alcance aquele ou aqueles, dependendo do problema proposto, que melhor atendam a certas condições previamente especificadas [Silva 2001]. Os AGs são recomendados para problemas NP-Completo e NP-Difíceis, devido ao tempo que AG leva para processar os dados [Linden 2008].

Para fazer a busca do melhor resultado o AG executa os seguintes operadores:

- Seleção: Um dos meios de fazer a seleção de indivíduos para o *crossover*, e o mais simples, é o método da roleta no qual cada indivíduo da população recebe uma fatia da roleta, essa fatia é determinada pela avaliação (*Fitness*) do indivíduo desta forma quem estiver mais próximo da solução do problema terá mais chances de ser sorteado. Em seguida é feito o sorteio aleatório, seria análogo ao "rodar a roleta", desses indivíduos os selecionados faram parte do operador de *crossover* e são colocados em um população auxiliar [Man et al. 2001].
- Cruzamento ou *Crossover*: O operador de cruzamento e de mutação são operadores chave do AG para que a busca seja bem sucedida, são os operadores mais

importantes do AG [Man et al. 2001]. Existem várias estratégias para esta operação será utilizada neste artigo a estratégia de cruzamento de um ponto, um ponto de cruzamento é escolhido aleatoriamente e a partir dele é feita a troca genética dos indivíduos escolhidos na seleção (pais) para a geração dos novos indivíduos (filhos). As informações genéticas anteriores a este ponto de um dos pais é ligado as informações posteriores do outro pai gerando a cada cruzamento dois novos filhos, que são colocados na população auxiliar [Rezende 2005].

- **Mutação:** O operador mutação é necessário para que a população seja diversificada, permite chegar mais rápido ao objetivo da busca e para fugir dos mínimos locais, alterando aleatoriamente um ou mais genes dos indivíduos gerados pelo cruzamento.
- **Reinserção:** Para saber quais indivíduos da população auxiliar será colocado na nova população é utilizado o operador de reinserção. Com a estratégia do elitismo pode haver um crescimento grande de super indivíduo na população, mas usando uma estratégia mais equilibrada parece melhorar o desempenho do AG, essa estratégia é chamada de melhores pais e filhos onde são escolhidos $N/2$ melhores indivíduos geradores (pais) e $N/2$ melhores indivíduos criados (Filhos) no qual são inseridos na população gerando uma nova geração [Man et al. 2001].

Em 2011 foi proposto pelos autores [Amaral and Jr. 2011] um algoritmo genético com um operador transgênico, esse novo operador para o algoritmo genético foi inspirado pela engenharia genética, onde permite manipular o material genético de indivíduos adicionando características que acredita-se ser a mais importante. Essa mesma atividade que é feita por engenheiros genéticos foi introduzido no AG, assim identificasse qual é o gene ou genes mais importante então esses genes dos N melhores indivíduos são passados para M outros indivíduos da população.

Um algoritmo genético, se utilizado para minerar dados, pode gerar regras e serem utilizadas em algum tipo de lógica. Lógica *Fuzzy* foi criada pois, havia algumas questões que a lógica booleana não conseguia afirmar com verdadeiro ou falso, então utilizando um grau de certeza que vai de 0-1 (0-100%) pode [Weber and Klein 2003].

O conceito *fuzzy* foi apresentado, em 1965, por Lotfi A. Zadeh [Zadeh 1965]; foi para ele o reconhecimento como colaborador dessa lógica e da inteligência artificial. porém o filósofo Platão já havia feito um estudo com uma terceira região de certeza, além do verdadeiro e falso [Weber and Klein 2003].

Em meados de 1960, Zadeh viu que recursos de automatização, de natureza industrial, biológica ou química, eram incapazes de compreender algumas situações ambíguas. Assim em 1965 o professor Zadeh publicou um artigo com os conceitos dos conjuntos *fuzzy* [Zadeh 1965] [Weber and Klein 2003].

A lógica *fuzzy* utiliza preposições linguísticas, exemplo Alto(x), Quente(x), Muito Rápido(x). Cada termo linguístico tem um valor de pertinência entre [0-1] [Ross 2010].

Existem três etapas na lógica *fuzzy* a primeira é a "fuzzificação", nessa etapa a aplicação difusa extrai os dados do mundo real e os transforma, por meio de uma função, em dados do universo difuso entre [0-1]. A segunda etapa é a inferência *fuzzy*, a aplicação difusa aplica as regras nos dados de entradas, já no universo [0-1], dado pela primeira etapa, e obtêm uma resposta. A terceira etapa é "defuzzificação", nessa etapa a aplicação

pega o resultado obtido pela regra que está no universo [0-1] e o transforma em uma saída com dados do mundo real por alguma função de "defuzzificação" [Ross 2010].

3. Solução Proposta

Para buscar um resultado favorável para o problema descrito neste artigo será utilizado algoritmo genético para minerar os dados e o conjunto *fuzzy* será usado para fazer a inferência do ambiente de tráfego do trecho, descrito na seção 1.

Será utilizado para fazer a mineração de dados o API Java do AG transgênico (AGT), criado pela Universidade Federal de Goiás *Campus* Jatai. Essa API (*Application Programming Interface*) segue os princípios descritos por [Holland 1975] e com o operador transgênico descrito por [Amaral and Jr. 2011]. O diagrama de atividades, figura 1, descreve como é o fluxo normal da API mostrada neste parágrafo.

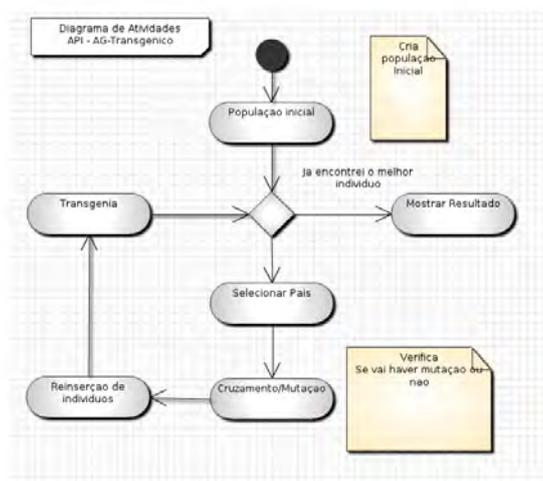


Figura 1. Diagrama de Atividades API AG-Transgênico

A configuração do cromossomo para o indivíduo do AG irá conter os genes: data (dia-mês), chuva com valores [Sim-Não], fluxo do tráfego valor inteiro e o congestionado, que será a classe para o minerador de dados, com valores [Sim-Não], o *fitness* ou avaliação do indivíduo se dá pela equação 1, os valores TN (*True Positive*, o minerador disse que era da classe 'Sim' e realmente era), FP (*False Positive*, o minerador disse que a classe não era 'Sim' e realmente não era) e os valores 0.5 relativos a 50% relativos aos dados [Sim,Não]. A possibilidade de existir uma mutação em algum indivíduo do AG será de 0,2%. O valor a ser buscado pelo AG é de um indivíduo com uma avaliação de 100%.

$$Fitness = TP * 0.5 + FP * 0.5; \quad (1)$$

A base de dados contém 25776 registros, que são os dados simulados que representa os três anos de coleta de dados dos pesquisadores como descrito na seção 1, a base apresenta uma quintupla que são os dados utilizados no cromossomo do indivíduo.

As regras difusas serão criadas de acordo com o resultado da mineração de dados levando em consideração o período e o fluxo do tráfego do trecho analisado e como termo lógicos serão:

Tabela 1. Resultados obtidos

Quantidade de testes	Itens Avaliados	AGT	AG
		Média	Média
50	Gerações	164	1250
	Fitness	100%	100%
	Tempo de Execução(Segundos)	43,553	87,341

Tabela 2. Linguísticas Para a Fuzzy

Conjuntos	valores de X[0-23]	valor de pertinência	Linguística
Período	$23 \geq X < 6$	$Y=0$	Baixo
	$6 \geq X < 12$	$Y=(N^{\circ} \text{ de registro do período } X \text{ na base de dados } *1)/1800$	Médio
	$13 > X < 17$		
	$18 > X < 23$		
	$12 \geq X \leq 13$	$Y=1$	Pico
$17 \geq X \leq 18$			
Fluxo	$0 \geq X \leq 20$	$Y=(X*1)/51$	Baixo
	$20 > X < 51$		Médio
	$X \leq 51$		Alto

- Período: Baixo, Médio e Pico;
- Fluxo: Baixo, Médio e Alto.

Logo após a criação das regras será usado um *software* para simular como será a reação do semáforo inteligente com a inferência *fuzzy*, o *software* irá obter as condições do trecho (Período e Fluxo) e com a ajuda das regras *fuzzy* tomar um decisão.

4. Resultados Obtidos

Para fazer uma análise de como o AG transgênico pode ser utilizado para encontrar padrões em grandes *data sets* foi executado 50 vezes, como mostra a Tabela 1. Em todos os testes o minerador de dados com AG chegou ao resultado de que o congestionamento ocorre quando o tráfego está acima ou igual a cinquenta e um. É perceptível que o algoritmo genético transgênico teve grandes resultados se comparado com o algoritmo genético tradicional, esse resultado pode ser justificado pelo fato do transgenico guiar a população do AG a um possível ótimo.

Com os resultados tragos pelo AG transgênico foi possível criar as fórmulas das linguísticas para a inferência *fuzzy*, essas fórmulas foram criadas pelos autores ao analisar a saída do minerador de dados, como mostra a Tabela 2. A saída da MD foi uma única regra que informava que classe congestionamento era sim se o valor de tráfego fosse ≥ 51 e se o valor fosse < 51 a classe seria não. As regras ficaram sendo as mostradas na Tabela 3, foi criada baseada na base de dados e na saída da mineração de dados, onde y é o grau de pertinência da entrada e s é o grau de pertinência da saída.

Foram testadas todas as possibilidades de ambiente e o simulador do semáforo respondeu corretamente, com forme as regras difusas, tendo assim uma resposta 100% para as inferências proposta neste artigo.

Tabela 3. Regras Nebulosas

Regras			
Rx	Período	Fluxo	Tempo Semáforo verde
R01	Medio(y) ou Pico(y)	Alto(y)	Alto(s)
R02	Baixo(y) ou Medio(y)	Medio(y)	Medio(s)
R03	Medio(y)	Baixo(y)	Medio(s)
R04	Baixo(y)	Baixo(y)	Baixo(s)

5. Conclusão e Trabalhos Futuros

Os resultados apresentados na seção anterior mostra que o AG Transgênico teve um desempenho de 50% melhor que o AG tradicional e o conjunto *Fuzzy*, baseado na mineração de dados, pode ser aplicado nos Semáforos de forma inteligente, onde o Semáforo toma sua decisão de acordo com o ambiente. Em todos os testes aplicados no simulador o Semáforo respondeu favorável o que demonstra que o objetivo do artigo foi válida.

Porém os testes foram feitos com um simulador utilizando dados reais e não foram feitos testes reais com o Semáforo, pois este artigo tem como objetivo mostrar a eficiência das inferências *fuzzy* com mineração de dados dentro de um problema da engenharia de tráfego. Para trabalho futuros pode ser feito uma melhor análise das regras. Utilizar mais de um Semáforo no teste para ver como seria a sincronização dos mesmos, será feita uma nova coleta de dados com novas variáveis para ser minerado e também, se possível, implementar em um semáforo real para ver como ele se comporta na prática.

Referências

- Amaral, L. R. (2007). Mineração de regras para classificação de oncogenes medidos por microarray utilizando algoritmos genéticos. Universidade Federal de Uberlândia. Dissertação de Mestrado.
- Amaral, L. R. and Jr., E. R. H. (2011). Transgenic, an operator for evolutionary algorithms. In *IEEE Congress on Evolutionary Computation*, pages 1308–1314. IEEE.
- Askerzade, I. N. and Mahmood, M. (2010). Control the extension time of traffic light in single junction by using fuzzy logic. *International Journal of Electrical e Computer Sciences*, 10(02):52–59.
- Bevilaqua, A., Rodrigues, F. A., and Amaral, L. R. (2011). Gasnp classifier: A machine learning environment for building high-level biological knowledge. *International Journal of Information Technology and Intelligent Computing*.
- Cox, E. (2005). *Fuzzy Modeling and Genetic Algorithms for Data Mining and Exploration*. The Morgan Kaufmann series in data management systems. Elsevier Science, USA, 1ª edição.
- Dantas, E. R. G., Junior, J. C. A. P., Lima, D. G., and Azevedo, R. R. (2008). O uso da descoberta de conhecimento em base de dados para apoiar a tomada de decisões. *V Simpósio de Excelência em Gestão e Tecnologia - SEGeT*, 01:50–60.
- Dias, R. A. (2004). Engenharia de tráfego em redes ip sobre tecnologia mpls: Otimização baseada em heurísticas. Universidade Federal de Santa Catarina, UFSC, Brasil. Tese de Doutorado.

- Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI, USA.
- Homes, G., Donkin, A., and Witten, I. H. (1994). Weka: A machine learning workbench. *Proceedings of the Second Australia and New Zealand Conference on Intelligent Information Systems*.
- Karakuzu, C. and Demirci, O. (2010). Fuzzy logic based smart traffic light simulator design and hardware implementation. *Appl. Soft Comput.*, 10(1):66–73.
- Kerner, B. S. (2009). *introduction to modern traffic flow theory and control the long road to three phase traffic theory*. Springer, Berlin, Alemanha.
- Lämmer, S. and Helbing, D. (2008). Self-control of traffic lights and vehicle flows in urban road networks. 0802.0403.
- Linden, R. (2008). *Algoritmos Genéticos: uma importante ferramenta da inteligencia computacional*, volume 3. Brasport Livros e Multimídia, São Paulo, Brasil.
- Man, K. F., Tang, K. S., and Kwong, S. (2001). *Genetic Algorithms: Concepts and Designs*. Spring, 3 edição.
- Minaei-Bidgoli, B. and Punch, W. F. (2003). Using genetic algorithms for data mining optimization in an educational web-based system. In *Proceedings of the 2003 international conference on Genetic and evolutionary computation: PartII*, GECCO'03, pages 2252–2263, Berlin, Heidelberg. Springer-Verlag.
- Prado, A. L. (2010). Modelagem matemática do fluxo de tráfego veicular. Universidade Federal do Paraná, UFPR, Brasil. Tese de Doutorado.
- Rezende, S. O. (2005). *Sistemas inteligentes: fundamentos e aplicações*. Manole Ltda, Barueri, SP.
- Ross, T. (2010). *Fuzzy logic with engineering applications*. John Wiley e Sons, Inglaterra, 3 edição.
- Shah, S. and Kusiak, A. (2007). Cancer gene search with data-mining and genetic algorithms. *Comput. Biol. Med.*, 37(2):251–261.
- Silva, E. E. (2001). Otimização de estruturas de concreto armado utilizando algoritmos genético. Escola Politécnica da Universidade de São Paulo. Dissertação de Mestrado.
- Tan, P.-N., Steinbach, M., and Kumar, V. (2009). *Introdução ao datamining*. Ciência Moderna, Rio de Janeiro, Brasil.
- Tonguz, O., Viriyasitavat, W., and Bai, F. (2009). Modeling urban traffic: A cellular automata approach. *Communications Magazine, IEEE*, 47(5):142–150.
- Weber, L. and Klein, P. A. T. (2003). *Aplicação da Lógica Fuzzy em Software e Hardware*. Ulbra, Rio Grande do Sul, Brasil.
- Yang, W., Zhang, L., He, Z., Yang, Y., and Fang, Y. (2012). Urban traffic signal two-stage combination fuzzy control and paramics simulation. In *Systems and Informatics (ICSAI), 2012 International Conference on*, pages 771–775.
- Zadeh, L. (1965). Fuzzy sets. *Information and Control*, 8(3):338–353.